Field Support Offices

EAST

*500 Office Center
Fort Washington Ind. Park
Fort Washington, Pennsylvania 19034
315/628-9050

*60 Turner Street
Waltham, Massachusettes 02154
817/899-9107

*Exchange Bank Building
1111 North Westshore Blvd
Suite 414
Tampa, Florida 33607
813/876-1304

*34 West Putnam
2nd Floor
Greenwich, Conn. 06830
203/622-0955

CENTRAL

*701 East Irving Park Blvd
Suite 206
Roselle, Illinois 60173
312/529-3993

5100 Edina Ind. Blvd
Edina, Minnesota 55436
612/835-7303

*228 Byers Road
Suite 105
Miamisburg, Ohio 45342
513/866-3405

WEST

*17870 Skypark Circle
Suite 107
Irvine, California 92714
714/549-0397

13300 Branchview Lane
Farmers Branch, Texas 75234
214/243-1017

*2025 Gateway Place
Suite 268
San Jose, California 95011
408/287-5081

9828 North Central Avenue
Suite 201
Phoenix, Arizona 85020
602/997-7573

*Offices that offer Field Application Engineering assistance.

Technical data is also available from your local MOSTEK representative or distributor. Please call!!

MOSTEK®
Z80·F8 Covering the full spectrum of 3870 microcomputer applications.

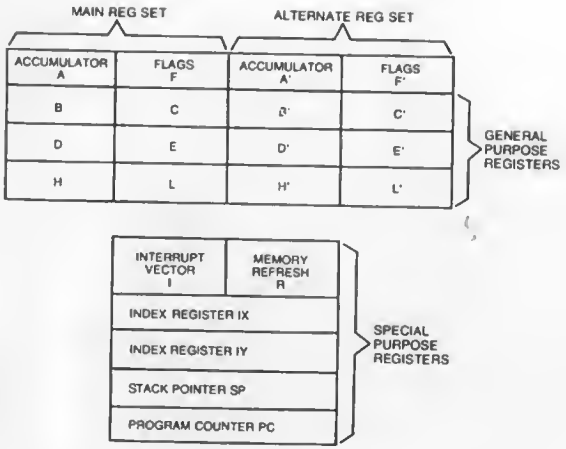
1215 W. Crosby Rd. • Carrollton, Texas 75006 • 214/242-0444
In Europe contact MOSTEK GmbH Talstrasse 172
D 7024 Filderstadt-1 W. Germany • Tele (0711) 701096

PRINTED IN USA February 1978
Publication No. MK78518

MOSTEK®

Z80 MICROCOMPUTER SYSTEM

Micro-Reference Manual



Z80-CPU REGISTER CONFIGURATION

SUMMARY OF FLAG OPERATION

Instruction	D7 S	Z		H		P/V	N	D0 C	Comments
ADD A, s; ADC A, s	↕	↕	X	↕	X	V	0	↕	8 bit add or add with carry
SUB s; SBC A, s; CP s; NEG	↕	↕	X	↕	X	V	1	↕	8 bit subtract, subtract with carry, compare and negate accumulator
AND s	↕	↕	X	1	X	P	0	0	Logical operations
OR s; XOR s	↕	↕	X	0	X	P	0	0	
INC s	↕	↕	X	↕	X	V	0	•	8 bit increment
DEC s	↕	↕	X	↕	X	V	1	•	8 bit decrement
ADD DD, SS	•	•	X	X	X	•	0	↕	16 bit add
ADC HL, SS	↕	↕	X	X	X	V	0	↕	16 bit add with carry
SBC HL, SS	↕	↕	X	X	X	V	1	↕	16 bit subtract with carry
RLA, RLCA, RRA, RRCA	•	•	X	0	X	•	0	↕	Rotate accumulator
RL s, RLC s, RR s, RRC s, SLA s, SRA s, SRL s	↕	↕	X	0	X	P	0	↕	Rotate and shift locations
RLD, RRD	↕	↕	X	0	X	P	0	•	Rotate digit left and right
DAA	↕	↕	X	↕	X	P	•	↕	Decimal adjust accumulator
CPL	•	•	X	1	X	•	1	•	Complements accumulator
SCF	•	•	X	0	X	•	0	1	Set carry
CCF	•	•	X	X	X	•	0	↕	Complement carry
IN r, (C)	↕	↕	X	0	X	P	0	•	Input register indirect
INI, IND, OUTI, OUTD	X	↕	X	X	X	X	1	•	Block input and output
INIR, INDR, OTIR, OTDR	X	1	X	X	X	X	1	•	Z = 0 if B ≠ 0 otherwise Z = 1
LDI, LDD	X	X	X	0	X	↕	0	•	Block transfer instructions
LDIR, LDDR	X	X	X	0	X	0	0	•	P/V = 1 if BC ≠ 0, otherwise P/V = 0
CPI, CPIR, CPD, CPDR	X	↕	X	X	X	↕	1	•	Block search instructions Z = 1 if A = (HL), otherwise Z = 0 P/V = 1 if BC ≠ 0, otherwise P/V = 0
LD A, I; LD A, R	↕	↕	X	0	X	IFF	0	•	The content of the interrupt enable flip flop (IFF) is copied into the P/V flag
BIT b, s	X	↕	X	1	X	X	0	•	The state of bit b of location s is copied into the Z flag

The following notation is used in this table

Symbol	Operation
C	Carry/link flag. C=1 if the operation produced a carry from the MSB of the operand or result.
Z	Zero flag. Z=1 if the result of the operation is zero.
S	Sign flag. S=1 if the MSB of the result is one.
P/V	Parity or overflow flag. Parity (P) and overflow (V) share the same flag. Logical operations affect this flag with the parity of the result while arithmetic operations affect this flag with the overflow of the result. If P/V holds parity, P/V=1 if the result of the operation is even, P/V=0 if result is odd. If P/V holds overflow, P/V=1 if the result of the operation produced an overflow.
H	Half carry flag. H=1 if the add or subtract operation produced a carry into or borrow from bit 4 of the accumulator.
N	Add/Subtract flag. N=1 if the previous operation was a subtract.
	H and N flags are used in conjunction with the decimal adjust instruction (DAA) to properly correct the result into packed BCD format following addition or subtraction using operands with packed BCD format.
↕	The flag is affected according to the result of the operation.
•	The flag is unchanged by the operation.
0	The flag is reset by the operation.
1	The flag is set by the operation.
X	The flag is a "don't care".
V	P/V flag affected according to the overflow result of the operation.
P	P/V flag affected according to the parity result of the operation.
r	Any one of the CPU registers A, B, C, D, E, H, L.
s	Any 8 bit location for all the addressing modes allowed for the particular instruction.
ss	Any 16 bit location for all the addressing modes allowed for that instruction.
ii	Any one of the two index registers IX or IY.
R	Refresh counter.
n	8 bit value in range <0, 255>.
nn	16 bit value in range <0, 65535>.

SOURCE

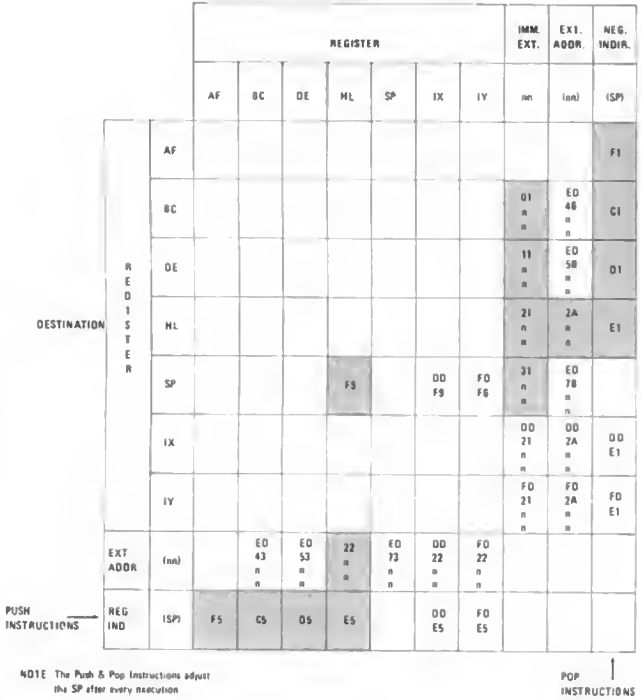
DESTINATION			IMPLIED		REGISTER							REG INDIRECT			INDEXED		EXT ADDR	IMME.
			I	R	A	B	C	D	E	H	L	(HL)	(BC)	(DE)	(IX+d)	(IY+d)	(nn)	n
	REGISTER	A	ED 57	ED 5F	7F	78	79	7A	7B	7C	7D	7E	0A	1A	DD 7E d	FD 7E d	3A n n	3E n
		B			47	40	41	42	43	44	45	46			DD 46 d	FD 46 d		06 n
		C			4F	48	49	4A	4B	4C	4D	4E			DD 4E d	FD 4E d		0E n
		D			57	50	51	52	53	54	55	56			DD 56 d	FD 56 d		16 n
		E			5F	58	59	5A	5B	5C	5D	5E			DD 5E d	FD 5E d		1E n
		H			67	60	61	62	63	64	65	66			DD 66 d	FD 66 d		26 n
		L			6F	68	69	6A	6B	6C	6D	6E			DD 6E d	FD 6E d		2E n
	REG INDIRECT	(HL)			77	70	71	72	73	74	75							36 n
		(BC)			02													
		(DE)			12													
	INDEXED	(IX+d)			DD 77 d	DD 70 d	DD 71 d	DD 72 d	DD 73 d	DD 74 d	DD 75 d							DD 36 d n
		(IY+d)			FD 77 d	FD 70 d	FD 71 d	FD 72 d	FD 73 d	FD 74 d	FD 75 d							FD 36 d n
	EXT.ADDR	(nn)			32 n n													
	IMPLIED	I			ED 47													
		R			ED 4F													

Mnemonic	Symbolic Operation	Flags S	Z		H		P/V	N	C	Op-Code 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments r, s	Reg.
LD r, s	r ← s	•	•	X	•	X	•	•	•	01 r s		1	1	4	r, s	Reg.
LD r, n	r ← n	•	•	X	•	X	•	•	•	00 r 110 ← n →		2	2	7	000	B
															001	C
LD r, (HL)	r ← (HL)	•	•	X	•	X	•	•	•	01 r 110		1	2	7	010	D
LD r, (IX+d)	r ← (IX+d)	•	•	X	•	X	•	•	•	11 011 101 01 r 110 ← d →	DD	3	5	15	011	E
															100	H
LD r, (IY+d)	r ← (IY+d)	•	•	X	•	X	•	•	•	11 111 101 01 r 110 ← d →	FD	3	5	19	101	L
															111	A
LD (HL), r	(HL) ← r	•	•	X	•	X	•	•	•	01 110 r		1	2	7		
LD (IX+d), r	(IX+d) ← r	•	•	X	•	X	•	•	•	11 011 101 01 110 r ← d →	DD	3	5	19		
LD (IY+d), r	(IY+d) ← r	•	•	X	•	X	•	•	•	11 111 101 01 110 r ← d →	FD	3	5	19		
LD (HL), n	(HL) ← n	•	•	X	•	X	•	•	•	00 110 110 ← n →	36	2	3	10		
LD (IX+d), n	(IX+d) ← n	•	•	X	•	X	•	•	•	11 011 101 00 110 110 ← d → ← n →	DD 36	4	5	19		
LD (IY+d), n	(IY+d) ← n	•	•	X	•	X	•	•	•	11 111 101 00 110 110 ← d → ← n →	FD 36	4	5	19		
LD A, (BC)	A ← (BC)	•	•	X	•	X	•	•	•	00 001 010	0A	1	2	7		
LD A, (DE)	A ← (DE)	•	•	X	•	X	•	•	•	00 011 010	1A	1	2	7		
LD A, (nn)	A ← (nn)	•	•	X	•	X	•	•	•	00 111 010 ← n → ← n →	3A	3	4	13		
LD (BC), A	(BC) ← A	•	•	X	•	X	•	•	•	00 000 010	02	1	2	7		
LD (DE), A	(DE) ← A	•	•	X	•	X	•	•	•	00 010 010	12	1	2	7		
LD (nn), A	(nn) ← A	•	•	X	•	X	•	•	•	00 110 010 ← n → ← n →	32	3	4	13		
LD A, I	A ← I	↕	↕	X	0	X	IFF	0	•	11 101 101 01 010 111	ED 57	2	2	9		
LD A, R	A ← R	↕	↕	X	0	X	IFF	0	•	11 101 101 01 011 111	ED 5F	2	2	9		
LD I, A	I ← A	•	•	X	•	X	•	•	•	11 101 101 01 000 111	ED 47	2	2	9		
LD R, A	R ← A	•	•	X	•	X	•	•	•	11 101 101 01 001 111	ED 4F	2	2	9		

Notes: r, s means any of the registers A, B, C, D, E, H, L
IFF the content of the interrupt enable flip-flop (IFF) is copied into the P/V flag

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,
↕ = flag is affected according to the result of the operation.

16-BIT LOAD GROUP 'LD' 'PUSH' AND 'POP'



4D1E The Push & Pop instructions adjust the SP after every execution

16-BIT LOAD GROUP

Mnemonic	Symbolic Operation	Flags S	Z		H		P/V	N	C	Op-Code 76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
LD dd, nn	dd ← nn	•	•	X	•	X	•	•	•	00	dd0	001		3	3	10	dd Pair
										←	n	→					00 BC
										←	n	→					01 DE
LD IX, nn	IX ← nn	•	•	X	•	X	•	•	•	11	011	101	DD	4	4	14	10 HL
										00	100	001	21				11 SP
										←	n	→					
										←	n	→					
LD IY, nn	IY ← nn	•	•	X	•	X	•	•	•	11	111	101	FD	4	4	14	
										00	100	001	21				
										←	n	→					
										←	n	→					
LD HL, (nn)	H ← (nn+1)	•	•	X	•	X	•	•	•	00	101	010	2A	3	5	16	
	L ← (nn)									←	n	→					
										←	n	→					
LD dd, (nn)	dd_H ← (nn+1)	•	•	X	•	X	•	•	•	11	101	101	ED	4	6	20	
	dd_L ← (nn)									01	dd1	011					
										←	n	→					
										←	n	→					
LD IX, (nn)	IX_H ← (nn+1)	•	•	X	•	X	•	•	•	11	011	101	DD	4	6	20	
	IX_L ← (nn)									00	101	010	2A				
										←	n	→					
										←	n	→					
LD IY, (nn)	IY_H ← (nn+1)	•	•	X	•	X	•	•	•	11	111	101	FD	4	6	20	
	IY_L ← (nn)									00	101	010	2A				
										←	n	→					
										←	n	→					
LD (nn), HL	(nn+1) ← H	•	•	X	•	X	•	•	•	00	100	010	22	3	5	16	
	(nn) ← L									←	n	→					
										←	n	→					
LD (nn), dd	(nn+1) ← dd_H	•	•	X	•	X	•	•	•	11	101	101	ED	4	6	20	
	(nn) ← dd_L									01	dd0	011					
										←	n	→					
										←	n	→					
LD (nn), IX	(nn+1) ← IX_H	•	•	X	•	X	•	•	•	11	011	101	DD	4	6	20	
	(nn) ← IX_L									00	100	010	22				
										←	n	→					
										←	n	→					
LD (nn), IY	(nn+1) ← IY_H	•	•	X	•	X	•	•	•	11	111	101	FD	4	6	20	
	(nn) ← IY_L									00	100	010	22				
										←	n	→					
										←	n	→					
LD SP, HL	SP ← HL	•	•	X	•	X	•	•	•	11	111	001	F9	1	1	6	
LD SP, IX	SP ← IX	•	•	X	•	X	•	•	•	11	011	101	DD	2	2	10	
										11	111	001	F9				
LD SP, IY	SP ← IY	•	•	X	•	X	•	•	•	11	111	101	FD	2	2	10	
										11	111	001	F9				qq Pair
PUSH qq	(SP-2) ← qq_L	•	•	X	•	X	•	•	•	11	qq0	101		1	3	11	00 BC
	(SP-1) ← qq_H																01 DE
PUSH IX	(SP-2) ← IX_L	•	•	X	•	X	•	•	•	11	011	101	DD	2	4	15	10 HL
	(SP-1) ← IX_H									11	100	101	E5				11 AF
PUSH IY	(SP-2) ← IY_L	•	•	X	•	X	•	•	•	11	111	101	FD	2	4	15	
	(SP-1) ← IY_H									11	100	101	E5				
POP qq	qq_H ← (SP+1)	•	•	X	•	X	•	•	•	11	qq0	001		1	3	10	
	qq_L ← (SP)																
POP IX	IX_H ← (SP+1)	•	•	X	•	X	•	•	•	11	011	101	DD	2	4	14	
	IX_L ← (SP)									11	100	001	E1				
POP IY	IY_H ← (SP+1)	•	•	X	•	X	•	•	•	11	111	101	FD	2	4	14	
	IY_L ← (SP)									11	100	001	E1				

Notes: dd is any of the register pairs BC, DE, HL, SP
qq is any of the register pairs AF, BC, DE, HL
$(PAIR)_H$, $(PAIR)_L$ refer to high order and low order eight bits of the register pair respectively.
e.g. BC_L = C, AF_H = A

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,
↕ flag is affected according to the result of the operation.

EXCHANGES 'EX' AND 'EXX'

		IMPLIED ADDRESSING				
		AF'	BC', DE' & HL'	HL	IX	IY
IMPLIED	AF	08				
	BC, DE & HL	D9				
	DE			EB		
REG. INDIR.	(SP)			E3	DD E3	FD E3

BLOCK TRANSFER GROUP

DESTINATION		SOURCE	
		REG. INDIR.	
		(HL)	
REG. INDIR.	(DE)	ED A0	'LDI' – Load (DE) ← (HL) Inc HL & DE, Dec BC
		ED B0	'LDIR' – Load (DE) ← (HL) Inc HL & DE, Dec BC, Repeat until BC = 0
		ED A8	'LDD' – Load (DE) ← (HL) Dec HL & DE, Dec BC
		ED B8	'LDDR' – Load (DE) ← (HL) Dec HL & DE, Dec BC, Repeat until BC = 0

HL points to source
DE points to destination
BC is byte counter

BLOCK SEARCH GROUP

SEARCH LOCATION	
REG INDIR	
(HL)	
ED A1	'CPI' Inc HL, Dec BC
ED B1	'CPIR' – Inc HL, Dec BC repeat until BC = 0 or find match
ED A9	'CPD' – Dec HL & BC
ED B9	'CPDR' – Dec HL & BC Repeat until BC = 0 or find match

HL points to location in memory to be compared with accumulator contents
BC is byte counter

EXCHANGE GROUP AND BLOCK TRANSFER AND SEARCH GROUP

Mnemonic	Symbolic Operation	Flags S	Z		H		P/V	N	C	Op Code 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
EX DE, HL	DE ↔ HL	•	•	X	•	X	•	•	•	11 101 011	EB	1	1	4	
EX AF, AF'	AF ↔ AF'	•	•	X	•	X	•	•	•	00 001 000	08	1	1	4	
EXX	(BC ↔ BC') (DE ↔ DE') (HL ↔ HL')	•	•	X	•	X	•	•	•	11 011 001	D9	1	1	4	Register bank and auxiliary register bank exchange
EX (SP), HL	H ↔ (SP+1) L ↔ (SP)	•	•	X	•	X	•	•	•	11 100 011	E3	1	5	18	
EX (SP), IX	IX_H ↔ (SP+1) IX_L ↔ (SP)	•	•	X	•	X	•	•	•	11 011 101 11 100 011	DD E3	2	6	23	
EX (SP), IY	IY_H ↔ (SP+1) IY_L ↔ (SP)	•	•	X	•	X	•	•	•	11 111 101 11 100 011	FD E3	2	6	23	
LDI	(DE) ← (HL) DE ← DE+1 HL ← HL+1 BC ← BC-1	•	•	X	0	X	↕ (1)	0	•	11 101 101 10 100 000	ED A0	2	4	16	Load (HL) into (DE), increment the pointers and decrement the byte counter (BC)
LDIR	(DE) ← (HL) DE ← DE+1 HL ← HL+1 BC ← BC-1 Repeat until BC = 0	•	•	X	0	X	0	0	•	11 101 101 10 110 000	ED B0	2 2	5 4	21 16	If BC ≠ 0 If BC = 0
LDD	(DE) ← (HL) DE ← DE-1 HL ← HL-1 BC ← BC-1	•	•	X	0	X	↕ (1)	0	•	11 101 101 10 101 000	ED A8	2	4	16	
LDDR	(DE) ← (HL) DE ← DE-1 HL ← HL-1 BC ← BC-1 Repeat until BC = 0	•	•	X	0	X	0	0	•	11 101 101 10 111 000	ED B8	2 2	5 4	21 16	If BC ≠ 0 If BC = 0
CPI	A – (HL) HL ← HL+1 BC ← BC-1	↕	↕ (2)	X	↕	X	↕ (1)	1	•	11 101 101 10 100 001	ED A1	2	4	16	
CPIR	A – (HL) HL ← HL+1 BC ← BC-1 Repeat until A = (HL) or BC = 0	↕	↕ (2)	X	↕	X	↕ (1)	1	•	11 101 101 10 110 001	ED B1	2 2	5 4	21 16	If BC ≠ 0 and A ≠ (HL) If BC = 0 or A = (HL)
CPD	A – (HL) HL ← HL-1 BC ← BC-1	↕	↕ (2)	X	↕	X	↕ (1)	1	•	11 101 101 10 101 001	ED A9	2	4	16	
CPDR	A – (HL) HL ← HL-1 BC ← BC-1 Repeat until A = (HL) or BC = 0	↕	↕ (2)	X	↕	X	↕ (1)	1	•	11 101 101 10 111 001	ED B9	2 2	5 4	21 16	If BC ≠ 0 and A ≠ (HL) If BC = 0 or A = (HL)

Notes: (1) P/V Flag is 0 if the result of BC - 1 = 0, otherwise P/V = 1
(2) Z flag is 1 if A = (HL), otherwise Z = 0

Flag Notation • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown, ↕ = flag is affected according to the result of the operation.

8-BIT ARITHMETIC AND LOGIC

SOURCE

	REGISTER ADDRESSING							REG. INDIR.	INDEXED		IMMED.
	A	B	C	D	E	H	L	(HL)	(IX+d)	(IY+d)	n
'ADD'	87	80	81	82	83	84	85	86	DD 86 d	FD 86 d	C6 n
ADD w CARRY 'ADC'	8F	88	89	8A	8B	8C	8D	8E	DD 8E d	FD 8E d	CE n
SUBTRACT 'SUB'	97	90	91	92	93	94	95	96	DD 96 d	FD 96 d	D6 n
SUB w CARRY 'SBC'	9F	98	99	9A	9B	9C	9D	9E	DD 9E d	FD 9E d	DE n
'AND'	A7	A0	A1	A2	A3	A4	A5	A6	DD A6 d	FD A6 d	E6 n
'XOR'	AF	A8	A9	AA	AB	AC	AD	AE	DD AE d	FD AE d	EE n
'OR'	B7	B0	B1	B2	B3	B4	B5	B6	DD B6 d	FD B6 d	F6 n
COMPARE 'CP'	BF	B8	B9	BA	BB	BC	BD	BE	DD BE d	FD BE d	FE n
INCREMENT 'INC'	3C	04	0C	14	1C	24	2C	34	DD 34 d	FD 34 d	
DECREMENT 'DEC'	3D	05	0D	15	1D	25	2D	35	DD 35 d	FD 35 d	

8-BIT ARITHMETIC AND LOGICAL GROUP

Mnemonic	Symbolic Operation	Flags S	Z		H		P/V	N	C	Op Code 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
ADD A, r	A ← A + r	↕	↕	X	↕	X	V	0	↕	10 000 r		1	1	4	r Reg 000 B 001 C 010 D
ADD A, n	A ← A + n	↕	↕	X	↕	X	V	0	↕	11 000 110 ← n →		2	2	7	
ADD A, (HL)	A ← A + (HL)	↕	↕	X	↕	X	V	0	↕	10 000 110		1	2	7	011 E 100 H 101 L 111 A
ADD A, (IX+d)	A ← A + (IX+d)	↕	↕	X	↕	X	V	0	↕	11 011 101 10 000 110 ← d →	DD	3	5	19	
ADD A, (IY+d)	A ← A + (IY+d)	↕	↕	X	↕	X	V	0	↕	11 111 101 10 000 110 ← d →	FD	3	5	19	
ADC A, s	A ← A + s + CY	↕	↕	X	↕	X	V	0	↕	001					s is any of r, n, (HL), (IX+d), (IY+d) as shown for ADD instruction. The indicated bits replace the 000 in the ADD set above
SUB s	A ← A − s	↕	↕	X	↕	X	V	1	↕	010					
SBC A, s	A ← A − s − CY	↕	↕	X	↕	X	V	1	↕	011					
AND s	A ← A ∧ s	↕	↕	X	1	X	P	0	0	100					
OR s	A ← A ∨ s	↕	↕	X	0	X	P	0	0	110					
XOR s	A ← A ⊕ s	↕	↕	X	0	X	P	0	0	101					
CP s	A − s	↕	↕	X	↕	X	V	1	↕	111					
INC r	r ← r + 1	↕	↕	X	↕	X	V	0	•	00 r 100		1	1	4	
INC (HL)	(HL) ← (HL) + 1	↕	↕	X	↕	X	V	0	•	00 110 100		1	3	11	
INC (IX+d)	(IX+d) ← (IX+d) + 1	↕	↕	X	↕	X	V	0	•	11 011 101 00 110 100 ← d →	DD	3	6	23	
INC (IY+d)	(IY+d) ← (IY+d) + 1	↕	↕	X	↕	X	V	0	•	11 111 101 00 110 100 ← d →	FD	3	6	23	
DEC s	s ← s − 1	↕	↕	X	↕	X	V	1	•	101					s is any of r, (HL), (IX+d), (IY+d) as shown for INC. DEC same format and states as INC. Replace 100 with 101 in OP Code

Notes: The V symbol in the P/V flag column indicates that the P/V flag contains the overflow of the result of the operation. Similarly the P symbol indicates parity. V = 1 means overflow, V = 0 means not overflow, P = 1 means parity of the result is even, P = 0 means parity of the result is odd.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown, ↕ = flag is affected according to the result of the operation

GENERAL PURPOSE AF OPERATIONS

Decimal Adjust Acc, 'DAA'	27
Complement Acc, 'CPL'	2F
Negate Acc, 'NEG' (2's complement)	ED 44
Complement Carry Flag, 'CCF'	3F
Set Carry Flag, 'SCF'	37

MISCELLANEOUS CPU CONTROL

'NOP'	00
'HALT'	76
DISABLE INT '(DI)'	F3
ENABLE INT '(EI)'	FB
SET INT MODE 0 'IM 0'	ED 46
SET INT MODE 1 'IM 1'	ED 56
SET INT MODE 2 'IM 2'	ED 5E

8080A MODE

RESTART TO LOCATION 0038_H

INDIRECT CALL USING REGISTER I AND 8 BITS FROM INTERRUPTING DEVICE AS A POINTER

GENERAL PURPOSE ARITHMETIC AND CPU CONTROL GROUPS

Mnemonic	Symbolic Operation	Flags S	Z		H		P/V	N	C	Op-Code 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
DAA	Converts acc. content into packed BCD following add or subtract with packed BCD operands	↕	↕	X	↕	X	P	•	↕	00 100 111	27	1	1	4	Decimal adjust accumulator
CPL	$A \leftarrow \overline{A}$	•	•	X	1	X	•	1	•	00 101 111	2F	1	1	4	Complement accumulator (One's complement)
NEG	$A \leftarrow \overline{A} + 1$	↕	↕	X	↕	X	V	1	↕	11 101 101 01 000 100	ED 44	2	2	8	Negate acc. (two's complement)
CCF	$CY \leftarrow \overline{CY}$	•	•	X	X	X	•	0	↕	00 111 111	3F	1	1	4	Complement carry flag
SCF	CY ← 1	•	•	X	0	X	•	0	1	00 110 111	37	1	1	4	Set carry flag
NOP	No operation	•	•	X	•	X	•	•	•	00 000 000	00	1	1	4	
HALT	CPU halted	•	•	X	•	X	•	•	•	01 110 110	76	1	1	4	
DI	IFF ← 0	•	•	X	•	X	•	•	•	11 110 011	F3	1	1	4	
EI	IFF ← 1	•	•	X	•	X	•	•	•	11 111 011	FB	1	1	4	
IM 0	Set interrupt mode 0	•	•	X	•	X	•	•	•	11 101 101 01 000 110	ED 46	2	2	8	
IM 1	Set interrupt mode 1	•	•	X	•	X	•	•	•	11 101 101 01 010 110	ED 56	2	2	8	
IM 2	Set interrupt mode 2	•	•	X	•	X	•	•	•	11 101 101 01 011 110	ED 5E	2	2	8	

Notes: IFF indicates the interrupt enable flip-flop
CY indicates the carry flip-flop

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,
↕ = flag is affected according to the result of the operation
● = Interrupts are not sampled at the end of EI or DI

16-BIT ARITHMETIC

SOURCE

DESTINATION			BC	DE	HL	SP	IX	IY
'ADD'	HL		09	19	29	39		
	IX		DD 09	DD 19		DD 39	DD 29	
	IY		FD 09	FD 19		FD 39		FD 29
ADD WITH CARRY AND SET FLAGS 'ADC'	HL		ED 4A	ED 5A	ED 6A	ED 7A		
SUB WITH CARRY AND SET FLAGS 'SBC'	HL		ED 42	ED 52	ED 62	ED 72		
INCREMENT 'INC'			03	13	23	33	DD 23	FD 23
DECREMENT 'DEC'			0B	1B	2B	3B	DD 2B	FD 2B

16-BIT ARITHMETIC GROUP

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	Op-Code 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
ADD HL, ss	HL ← HL+ss	•	•	X	X	•	0	‡	00 ss1 001		1	3	11	ss Reg.
														00 BC
ADC HL, ss	HL ← HL+ss+CY	‡	‡	X	X	V	0	‡	11 101 101	ED	2	4	15	01 DE
									01 ss1 010					10 HL
														11 SP
SBC HL, ss	HL ← HL-ss-CY	‡	‡	X	X	V	1	‡	11 101 101	ED	2	4	15	
									01 ss0 010					
ADD IX, pp	IX ← IX + pp	•	•	X	X	•	0	‡	11 011 101	DD	2	4	15	pp Reg
									00 pp1 001					00 BC
														01 DE
														10 IX
														11 SP
ADD IY, rr	IY ← IY + rr	•	•	X	X	•	0	‡	11 111 101	FD	2	4	15	rr Reg.
									00 rr1 001					00 BC
														01 DE
														10 IY
														11 SP
INC ss	ss ← ss + 1	•	•	X	•	X	•	•	00 ss0 011		1	1	6	
INC IX	IX ← IX + 1	•	•	X	•	X	•	•	11 011 101	DD	2	2	10	
									00 100 011	23				
INC IY	IY ← IY + 1	•	•	X	•	X	•	•	11 111 101	FD	2	2	10	
									00 100 011	23				
DEC ss	ss ← ss 1	•	•	X	•	X	•	•	00 ss1 011		1	1	6	
DEC IX	IX ← IX 1	•	•	X	•	X	•	•	11 011 101	DD	2	2	10	
									00 101 011	2B				
DEC IY	IY ← IY 1	•	•	X	•	X	•	•	11 111 101	FD	2	2	10	
									00 101 011	2B				

Notes: ss is any of the register pairs BC, DE, HL, SP
pp is any of the register pairs BC, DE, IX, SP
rr is any of the register pairs BC, DE, IY, SP

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown.
‡ = flag is affected according to the result of the operation.

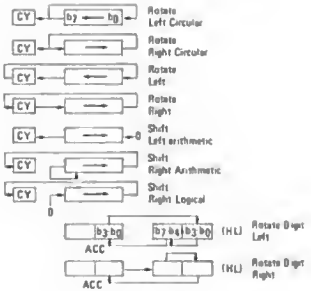
ROTATES AND SHIFTS

Source and Destination

TYPE OF ROTATE OR SHIFT

	A	B	C	D	E	H	L	(HL)	(IX+d)	(IY+d)
'RLC'	CB 07	CB 00	CB 01	CB 02	CB 03	CB 04	CB 05	CB 06	DD CB d 06	FD CB d 06
'RRC'	CB 0F	CB 08	CB 09	CB 0A	CB 0B	CB 0C	CB 0D	CB 0E	DD CB d 0E	FD CB d 0E
'RL'	CB 17	CB 10	CB 11	CB 12	CB 13	CB 14	CB 15	CB 16	DD CB d 16	FD CB d 16
'RR'	CB 1F	CB 18	CB 19	CB 1A	CB 1B	CB 1C	CB 1D	CB 1E	DD CB d 1E	FD CB d 1E
'SLA'	CB 27	CB 20	CB 21	CB 22	CB 23	CB 24	CB 25	CB 26	DD CB d 26	FD CB d 26
'SRA'	CB 2F	CB 28	CB 29	CB 2A	CB 2B	CB 2C	CB 2D	CB 2E	DD CB d 2E	FD CB d 2E
'SRL'	CB 3F	CB 38	CB 39	CB 3A	CB 3B	CB 3C	CB 3D	CB 3E	DD CB d 3E	FD CB d 3E
'RLD'								ED 6F		
'RRD'								ED 67		

	A
'RLCA'	07
'RRCA'	0F
'RLA'	17
'RRA'	1F



ROTATE AND SHIFT GROUP

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	Op-Code 7 6 5 4 3 2 1 0	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
RLCA		•	•	X	0	X	•	0	0	00 000 111	07	1	1	4	Rotate left circular accumulator
RLA		•	•	X	0	X	•	0	↕	00 010 111	17	1	1	4	Rotate left accumulator
RRCA		•	•	X	0	X	•	0	↕	00 001 111	0F	1	1	4	Rotate right circular accumulator
RRA		•	•	X	0	X	•	0	↕	00 011 111	1F	1	1	4	Rotate right accumulator
RLC r		↕	↕	X	0	X	P	0	↕	11 001 011 00 000 r	CB	2	2	8	Rotate left circular register r
RLC (HL)		↕	↕	X	0	X	P	0	↕	11 001 011 00 000 110	CB	2	4	15	r: 000 B, 001 C, 010 D, 011 E, 100 H, 101 L, 111 A
RLC (IX+d)	CY ← 7 ← 0 ← CY r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	11 011 101 11 001 011 – d – 00 000 110	DD CB	4	6	23	
RLC (IY+d)		↕	↕	X	0	X	P	0	↕	11 111 101 11 001 011 – d – 00 000 110	FD CB	4	6	23	
RL s	CY ← 7 ← 0 ← CY s = r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	010					Instruction format and states are as shown for RLC's. To form new Op-Code replace 000 of RLC's with shown code
RRC s	0 → 7 → 0 → CY s = r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	001					
RR s	0 → 7 → 0 → CY s = r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	011					
SLA s	CY ← 7 ← 0 ← 0 s = r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	100					
SRA s	7 → 0 → CY s = r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	101					
SRL s	0 → 7 → 0 → CY s = r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	111					
RLD	A 7-4 3-0 7-4 3-0 (HL)	↕	↕	X	0	X	P	0	•	11 101 101 01 101 111	ED 6F	2	5	18	Rotate digit left and right between the accumulator and location (HL). The content of the upper half of the accumulator is unaffected
RRD	A 7-4 3-0 7-4 3-0 (HL)	↕	↕	X	0	X	P	0	•	11 101 101 01 100 111	ED 67	2	5	18	

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown, ↕ = flag is affected according to the result of the operation.

BIT MANIPULATION GROUP

	BIT	REGISTER ADDRESSING							REG. INDIR.	INDEXED	
		A	B	C	D	E	H	L	(HL)	(IX+d)	(IY+d)
TEST 'BIT'	0	CB 47	CB 40	CB 41	CB 42	CB 43	CB 44	CB 45	CB 46	DD CB d 46	FD CB d 46
	1	CB 4F	CB 48	CB 49	CB 4A	CB 4B	CB 4C	CB 4D	CB 4E	DD CB d 4E	FD CB d 4E
	2	CB 57	CB 50	CB 51	CB 52	CB 53	CB 54	CB 55	CB 56	DD CB d 56	FD CB d 56
	3	CB 5F	CB 58	CB 59	CB 5A	CB 5B	CB 5C	CB 5D	CB 5E	DD CB d 5E	FD CB d 5E
	4	CB 67	CB 60	CB 61	CB 62	CB 63	CB 64	CB 65	CB 66	DD CB d 66	FD CB d 66
	5	CB 6F	CB 68	CB 69	CB 6A	CB 6B	CB 6C	CB 6D	CB 6E	DD CB d 6E	FD CB d 6E
	6	CB 77	CB 70	CB 71	CB 72	CB 73	CB 74	CB 75	CB 76	DD CB d 76	FD CB d 76
	7	CB 7F	CB 78	CB 79	CB 7A	CB 7B	CB 7C	CB 7D	CB 7E	DD CB d 7E	FD CB d 7E
RESET BIT 'RES'	0	CB 87	CB 80	CB 81	CB 82	CB 83	CB 84	CB 85	CB 86	DD CB d 86	FD CB d 86
	1	CB 8F	CB 88	CB 89	CB 8A	CB 8B	CB 8C	CB 8D	CB 8E	DD CB d 8E	FD CB d 8E
	2	CB 97	CB 90	CB 91	CB 92	CB 93	CB 94	CB 95	CB 96	DD CB d 96	FD CB d 96
	3	CB 9F	CB 98	CB 99	CB 9A	CB 9B	CB 9C	CB 9D	CB 9E	DD CB d 9E	FD CB d 9E
	4	CB A7	CB A0	CB A1	CB A2	CB A3	CB A4	CB A5	CB A6	DD CB d A6	FD CB d A6
	5	CB AF	CB A8	CB A9	CB AA	CB AB	CB AC	CB AD	CB AE	DD CB d AE	FD CB d AE
	6	CB B7	CB B0	CB B1	CB B2	CB B3	CB B4	CB B5	CB B6	DD CB d B6	FD CB d B6
	7	CB BF	CB B8	CB B9	CB BA	CB BB	CB BC	CB BD	CB BE	DD CB d BE	FD CB d BE
SET BIT 'SET'	0	CB C7	CB C0	CB C1	CB C2	CB C3	CB C4	CB C5	CB C6	DD CB d C6	FD CB d C6
	1	CB CF	CB C8	CB C9	CB CA	CB CB	CB CC	CB CD	CB CE	DD CB d CE	FD CB d CE
	2	CB D7	CB D0	CB D1	CB D2	CB D3	CB D4	CB D5	CB D6	DD CB d D6	FD CB d D6
	3	CB DF	CB D8	CB D9	CB DA	CB DB	CB DC	CB DD	CB DE	DD CB d DE	FD CB d DE
	4	CB E7	CB E0	CB E1	CB E2	CB E3	CB E4	CB E5	CB E6	DD CB d E6	FD CB d E6
	5	CB EF	CB E8	CB E9	CB EA	CB EB	CB EC	CB ED	CB EE	DD CB d EE	FD CB d EE
	6	CB F7	CB F0	CB F1	CB F2	CB F3	CB F4	CB F5	CB F6	DD CB d F6	FD CB d F6
	7	CB FF	CB F8	CB F9	CB FA	CB FB	CB FC	CB FD	CB FE	DD CB d FE	FD CB d FE

BIT SET, RESET AND TEST GROUP

Mnemonic	Symbolic Operation	Flags S	Z	H	P/V	N	C	Op Code 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
BIT b, r	$Z = \overline{r_b}$	X	‡	1	X	0	•	11 001 011 01 b r	CB	2	2	8	r: 000 B, 001 C, 010 D, 011 E, 100 H, 101 L, 111 A
BIT b, (HL)	$Z = \overline{(HL)_b}$	X	‡	1	X	0	•	11 001 011 01 b 110	CB	2	3	12	b: 000 Bit 0, 001 1, 010 2, 011 3, 100 4, 101 5, 110 6, 111 7
BIT b, (IX+d)	$Z = \overline{(IX+d)_b}$	X	‡	1	X	0	•	11 011 101 11 001 011 - d - 01 b 110	DD CB	4	5	20	
BIT b, (IY+d)	$Z = \overline{(IY+d)_b}$	X	‡	1	X	0	•	11 111 101 11 001 011 - d - 01 b 110	FD CB	4	5	20	
SET b, r	$r_b = 1$	•	•	X	•	X	•	•	11 001 011 11 b r	CB	2	2	8	
SET b, (HL)	$(HL)_b = 1$	•	•	X	•	X	•	•	11 001 011 11 b 110	CB	2	4	15	
SET b, (IX+d)	$(IX+d)_b = 1$	•	•	X	•	X	•	•	11 011 101 11 001 011 - d - 11 b 110	DD CB	4	6	23	
SET b, (IY+d)	$(IY+d)_b = 1$	•	•	X	•	X	•	•	11 111 101 11 001 011 - d - 11 b 110	FD CB	4	6	23	
RES b, s	$s_b = 0$ s ≡ r, (HL), (IX+d), (IY+d)	•	•	X	•	X	•	•	10				To form new Op Code replace 11 of SET b, s with 10. Flags and time states for SET instruction

Notes: The notation s_b indicates bit b (0 to 7) of location s.

Flag Notation: • = flag not effected, 0 = flag reset, 1 = flag set, X = flag is unknown, ‡ = flag is affected according to the result of the operation.

JUMP GROUP

CONDITION

			UN COND	CARRY	NON CARRY	ZERO	NON ZERO	PARITY EVEN	PARITY ODD	SIGN NEG	SIGN POS	REG B=0
JUMP "JP"	IMMED. EXT	nn	C3 n n	DA n n	D2 n n	CA n n	C2 n n	EA n n	E2 n n	FA n n	F2 n n	
JUMP "JR"	RELATIVE	PC + e	18 e-2	38 e-2	30 e-2	28 e-2	20 e-2					
JUMP "JP"	REG INDIR	(HL)	E9									
JUMP "JP"		(IX)	DD E9									
JUMP "JP"		(IY)	FD E9									
DECREMENT B, JUMP IF NON ZERO "DJNZ"	RELATIVE	PC + e										10 e-2

JUMP GROUP

Mnemonic	Symbolic Operation	Flags S	Z	H	P/V	N	C	Op-Code 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
JP nn	PC ← nn	•	•	X	•	X	•	•	•	11 000 011 ← n → ← n →	C3	3	3	10	
JP cc, nn	If condition cc is true PC ← nn, otherwise continue	•	•	X	•	X	•	•	•	11 cc 010 ← n → ← n →		3	3	10	
JR e	PC ← PC + e	•	•	X	•	X	•	•	•	00 011 000 ← e-2 →	18	2	3	12	
JR C, e	If C = 0, continue	•	•	X	•	X	•	•	•	00 111 000 ← e-2 →	38	2	2	7	If condition not met
	If C = 1, PC ← PC+e											2	3	12	If condition is met
JR NC, e	If C = 1, continue	•	•	X	•	X	•	•	•	00 110 000 ← e-2 →	30	2	2	7	If condition not met
	If C = 0, PC ← PC+e											2	3	12	If condition is met
JR Z, e	If Z = 0 continue	•	•	X	•	X	•	•	•	00 101 000 ← e-2 →	28	2	2	7	If condition not met
	If Z = 1, PC ← PC+e											2	3	12	If condition is met
JR NZ, e	If Z = 1, continue	•	•	X	•	X	•	•	•	00 100 000 ← e-2 →	20	2	2	7	If condition not met
	If Z = 0, PC ← PC+e											2	3	12	If condition is met
JP (HL)	PC ← HL	•	•	X	•	X	•	•	•	11 101 001	E9	1	1	4	
JP (IX)	PC ← IX	•	•	X	•	X	•	•	•	11 011 101 11 101 001	DD E9	2	2	8	
JP (IY)	PC ← IY	•	•	X	•	X	•	•	•	11 111 101 11 101 001	FD E9	2	2	8	
DJNZ, e	B ← B-1 If B = 0, continue	•	•	X	•	X	•	•	•	00 010 000 ← e-2 →	10	2	2	8	If B = 0
	If B ≠ 0, PC ← PC+e											2	3	13	If B ≠ 0

cc	Condition
000	NZ non zero
001	Z zero
010	NC non carry
011	C carry
100	PO parity odd
101	PE parity even
110	P sign positive
111	M sign negative

Notes: e represents the extension in the relative addressing mode.

e is a signed two's complement number in the range <126, 129>

e-2 in the op-code provides an effective address of pc+e as PC is incremented by 2 prior to the addition of e.

Flag Notation. • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown, 1 = flag is affected according to the result of the operation.

CALL AND RETURN GROUP

			CONDITION									
			UN-COND.	CARRY	NOR CARRY	ZERO	NOR ZERO	PARITY EVEN	PARITY ODD	SIGN NEG.	SIGN POS.	REG. B ≠ 0
'CALL'	IMMED. EXT.	nn	C0 n n	DC n n	D4 n n	CC n n	C4 n n	EC n n	E4 n n	FC n n	F4 n n	
RETURN 'RET'	REGISTER INDIR.	(SP) (SP+1)	C9	D8	00	C8	C0	E8	E0	F8	F0	
RETURN FROM INT 'RETI'	REGISTER INDIR.	(SP) (SP+1)	ED 4D									
RETURN FROM NOR MASKABLE INT 'RETN'	REGISTER INDIR.	(SP) (SP+1)	ED 45									

NOTE – CERTAIN FLAGS HAVE MORE THAN ONE PURPOSE. REFER TO Z80 CPU TECHNICAL MANUAL FOR DETAILS.

RESTART GROUP

		OP CODE	
CALL ADDRESS	0000_H	C7	'RST 0'
	0008_H	CF	'RST 8'
	0010_H	D7	'RST 10'
	0018_H	DF	'RST 24'
	0020_H	E7	'RST 32'
	0028_H	EF	'RST 40'
	0030_H	F7	'RST 48'
	0038_H	FF	'RST 56'

CALL AND RETURN GROUP

Mnemonic	Symbolic Operation	S	Z		N		P/V	N	C	Op-Code 76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
CALL nn	(SP-1) ← PC_H (SP-2) ← PC_L PC ← nn	•	•	X	•	X	•	•	•	11 – –	001 n n	101 – –	CD	3	5	17	
CALL cc, nn	If condition cc is false continue, otherwise same as CALL nn	•	•	X	•	X	•	•	•	11 – –	cc n n	100 – –		3	3	10	If cc is false
														3	5	17	If cc is true
RET	PC_L ← (SP) PC_H ← (SP+1)	•	•	X	•	X	•	•	•	11	001	001	C9	1	3	10	
RET cc	If condition cc is false continue, otherwise same as RET	•	•	X	•	X	•	•	•	11	cc	000		1	1	5	If cc is false
														1	3	11	If cc is true
RETI	Return from interrupt	•	•	X	•	X	•	•	•	11 01	101 001	101 101	ED 4D	2	4	14	
RETN[1]	Return from non maskable interrupt	•	•	X	•	X	•	•	•	11 01	101 000	101 101	ED 45	2	4	14	
RST p	(SP-1) ← PC_H (SP-2) ← PC_L PC_H ← 0 PC_L ← p	•	•	X	•	X	•	•	•	11	t	111		1	3	11	

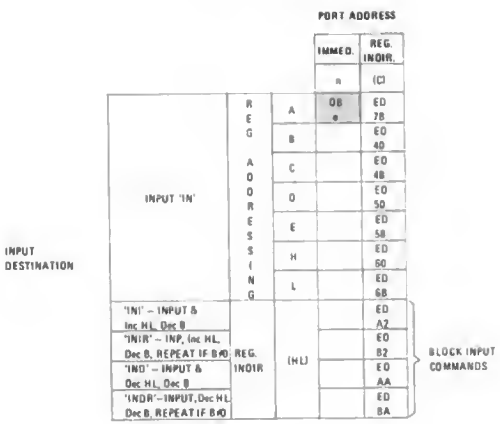
cc	Condition	
000	NZ	non zero
001	Z	zero
010	NC	no carry
011	C	carry
100	PO	parity odd
101	PE	parity even
110	P	sign positive
111	M	sign negative

t	p
000	00H
001	08H
010	10H
011	18H
100	20H
101	28H
110	30H
111	38H

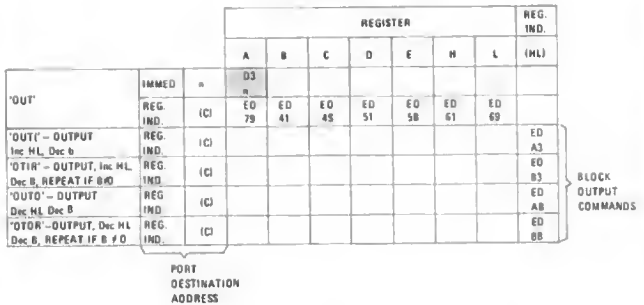
[1] RETN loads IFF_2 → IFF_1

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown, ↕ = flag is affected according to the result of the operation.

INPUT GROUP



OUTPUT GROUP



INPUT AND OUTPUT GROUP

Mnemonic	Symbolic Operation	Flags S	Z		H		P/V	N	C	Op-Code 76 543 210	Hex	No of Bytes	No. of M Cycles	No. of T States	Comments
IN A, (n)	A ← (n)	•	•	X	•	X	•	•	•	11 011 011 – n –	DB	2	3	11	n to A_0 ~ A_7 Acc to A_8 ~ A_{15}
IN r, (C)	r ← (C) if r = 110 only the flags will be affected	‡	‡	X	‡	X	P	0	•	11 101 101 01 r 000	ED	2	3	12	C to A_0 ~ A_7 B to A_8 ~ A_{15}
INI	(HL) ← (C) B ← B – 1 HL ← HL + 1	X	‡ (1)	X	X	X	X	1	•	11 101 101 10 100 010	ED A2	2	4	16	C to A_0 ~ A_7 B to A_8 ~ A_{15}
INIR	(HL) ← (C) B ← B – 1 HL ← HL + 1 Repeat until B = 0	X	1	X	X	X	X	1	•	11 101 101 10 110 010	ED B2	2 2	5 (if B ≠ 0) 4 (if B = 0)	21 16	C to A_0 ~ A_7 B to A_8 ~ A_{15}
IND	(HL) ← (C) B ← B – 1 HL ← HL – 1	X	‡ (1)	X	X	X	X	1	•	11 101 101 10 101 010	ED AA	2	4	16	C to A_0 ~ A_7 B to A_8 ~ A_{15}
INDR	(HL) ← (C) B ← B – 1 HL ← HL – 1 Repeat until B = 0	X	1	X	X	X	X	1	•	11 101 101 10 111 010	ED BA	2 2	5 (if B ≠ 0) 4 (if B = 0)	21 16	C to A_0 ~ A_7 B to A_8 ~ A_{15}
OUT (n), A	(n) ← A	•	•	X	•	X	•	•	•	11 010 011 – n –	D3	2	3	11	n to A_0 ~ A_7 Acc to A_8 ~ A_{15}
OUT (C), r	(C) ← r	•	•	X	•	X	•	•	•	11 101 101 01 r 001	ED	2	3	12	C to A_0 ~ A_7 B to A_8 ~ A_{15}
OUTI	(C) ← (HL) B ← B – 1 HL ← HL + 1	X	‡ (1)	X	X	X	X	1	•	11 101 101 10 100 011	ED A3	2	4	16	C to A_0 ~ A_7 B to A_8 ~ A_{15}
OTIR	(C) ← (HL) B ← B – 1 HL ← HL + 1 Repeat until B = 0	X	1	X	X	X	X	1	•	11 101 101 10 110 011	ED B3	2 2	5 (if B ≠ 0) 4 (if B = 0)	21 16	C to A_0 ~ A_7 B to A_8 ~ A_{15}
OUTD	(C) ← (HL) B ← B – 1 HL ← HL – 1	X	‡ (1)	X	X	X	X	1	•	11 101 101 10 101 011	ED AB	2	4	16	C to A_0 ~ A_7 B to A_8 ~ A_{15}
OTDR	(C) ← (HL) B ← B – 1 HL ← HL – 1 Repeat until B = 0	X	1	X	X	X	X	1	•	11 101 101 10 111 011	ED BB	2 2	5 (if B ≠ 0) 4 (if B = 0)	21 16	C to A_0 ~ A_7 B to A_8 ~ A_{15}

Notes: (1) H the result of B – 1 is zero the Z flag is set, otherwise it is reset

Flag Notation • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown, ‡ = flag is affected according to the result of the operation.

Z80 – CPU INTERRUPT STRUCTURE

MASKABLE ($\overline{INT}$)

Mode 0

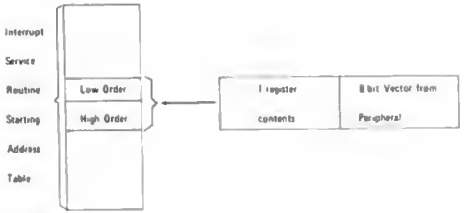
Place instruction onto Data Bus during $\overline{INTA} \cdot \overline{M1} \cdot \overline{IORQ}$ like 8080A

Mode 1

Restart to 38_H or 56_{10} (RST 56?)

Mode 2

Used by Z80 Peripherals



NON MASKABLE ($\overline{NMI}$)

Restart to 66_H or 102_{10}

INTERRUPT ENABLE/DISABLE FLIP FLOPS

Action	IFF_1	IFF_2	
CPU Reset	0	0	
DI	0	0	
EI	1	1	
LD A, I	•	•	IFF_2 = Parity flag
LD A, R	•	•	IFF_2 = Parity flag
Accept $\overline{NMI}$	0	•	
RETN	IFF_2	•	IFF_2 = IFF_1
Accept $\overline{INT}$	0	0	
RETI	•	•	

"•" indicates no change

PIO PROGRAMMING SUMMARY

REGISTER SELECTION

SELECT LINES C/D	SELECT LINES B/A	REGISTER SELECTED
0	0	A Data
0	1	B Data
1	0	A Control
1	1	B Control

LOAD INTERRUPT VECTOR

D7							D0	
V7	V6	V5	V4	V3	V2	V1	0	Control Register

SET OPERATING MODE

D7							D0	
M1	M0	X	X	1	1	1	1	Control Register

Mode Number	M1	M0	Mode
0	0	0	Output
1	0	1	Input
2	1	0	Bidirectional
3	1	1	Bit Control

If Mode 3 selected, the next control word to the PIO is

D7							D0	
I/O_7	I/O_6	I/O_5	I/O_4	I/O_3	I/O_2	I/O_1	I/O_0	Control Register

I/O = 1 Sets bit to Input

I/O = 0 Sets bit to Output

SET INTERRUPT CONTROL

D7							D0	
Int Enable	AND/OR	High/Low	Mask Follows	0	1	1	1	Control Register

In Mode 3 if Mask follows = 1, the next control word to the PIO is

D7							D0	
MB_7	MB_6	MB_5	MB_4	MB_3	MB_2	MB_1	MB_0	Control Register

MB = 0 Monitor the bit

MB = 1 Mask the bit

ENABLE / DISABLE INTERRUPTS

D7							D0	
Int Enable	X	X	X	0	0	1	1	Control Register

CTC PROGRAMMING SUMMARY

REGISTER SELECTION

SELECT LINES CS_1	SELECT LINES CS_0	CHANNEL SELECTED	PRIORITY
0	0	0	Highest
0	1	1	
1	0	2	
1	1	3	Lowest

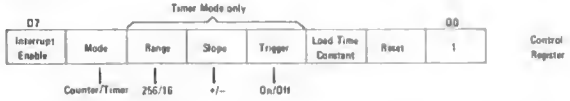
READ = DOWN COUNTER WRITE = CONTROL REGISTER

LOAD INTERRUPT VECTOR

$CS_0 = CS_1 = 0$

XX is the binary equivalent of interrupting channel number

SET OPERATING MODE



If Load Time Constant = 1 the next control word is the Time Constant:

CTC Channel interrupts when 01_H is decremented to 00_H

Time Content	Decimal counts to interrupt
01_H	1
•	•
•	•
FF_H	255
00_H	256

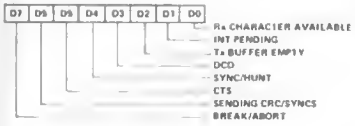
SIO PROGRAMMING SUMMARY

CHANNEL SELECTION

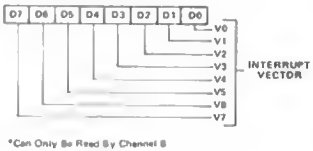
C/D	B/A	FUNCTION
0	0	Channel A Data
0	1	Channel B Data
1	0	Channel A Commands/Status
1	1	Channel B Commands/Status

READ REGISTERS

READ REGISTER 0

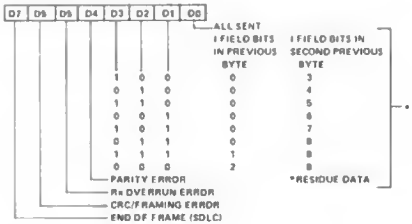


READ REGISTER 2*



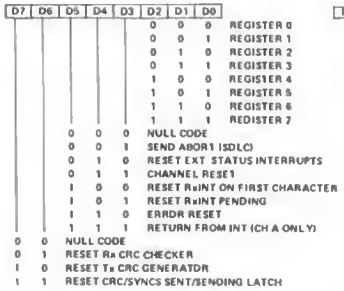
*Can Only Be Read By Channel B

READ REGISTER 1

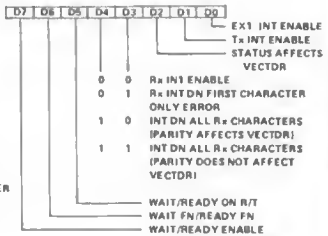


WRITE REGISTERS

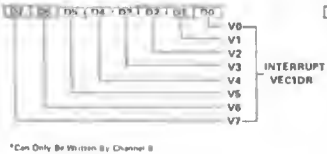
WRITE REGISTER 0



WRITE REGISTER 1

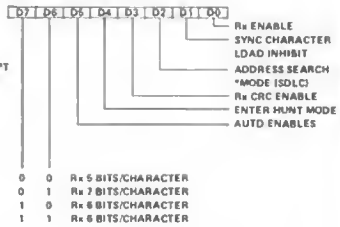


WRITE REGISTER 2*

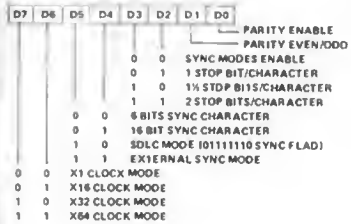


*Can Only Be Written By Channel B

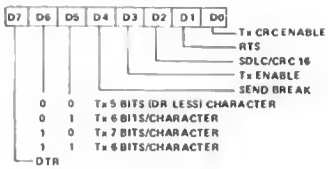
WRITE REGISTER 3



WRITE REGISTER 4



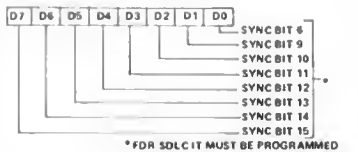
WRITE REGISTER 5



WRITE REGISTER 6

*ALSO SDLC ADDRESS FIELD

WRITE REGISTER 7



*FOR SDLC IT MUST BE PROGRAMMED TO "01111110" FOR FLAG RECOGNITION

STATUS AFFECTS VECTOR (D2) (FROM WRITE REG 1)

If this mode is selected, the vector returned from an interrupt acknowledge cycle will be variable according to the following

	V_3	V_2	V_1	
Ch B	0	0	0	Ch B Transmit Buffer Empty
	0	0	1	Ch B External/Status Change
	0	1	0	Ch B Receive Character Available
	0	1	1	Ch B Special Receive Condition
Ch A	1	0	0	Ch A Transmit Buffer Empty
	1	0	1	Ch A External/Status Change
	1	1	0	Ch A Receive Character Available
	1	1	1	Ch A Special Receive Condition

If this bit is 0, the fixed vector programmed in the vector register is returned.